MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2022/2023 (TERM ID 2215)

ERC1036 – COMPUTER & PROGRAMMING

(All Sections / Groups)

20 FEBRUARY 2023
2:30 P.M. – 4:30 P.M.
(2 hours)

INSTRUCTIONS TO STUDENT

1. This question paper consists of 11 printed pages including cover page with 4 questions only.

2. Answer ALL the questions. All questions carry equal marks and the distribution of marks for each question is given.

3. Please write all your answers in the answer booklet provided.

Question 1

(a) You are making a guess-the-number game for schoolchildren. They must guess the secret number in 3 attempts to win a prize. Assume that the secret number has been hard-coded to 24 by the programmer. The program will do the following:

- Display "Too big" if guess is larger than the secret number,
- Display "Too small" if guess is smaller than the secret number,
- Display "You guessed it!" if the guess is correct,
- Display "You lost!" after 3 incorrect attempts.

The partially written program is shown in Figure 1(a). Complete the rest of the program so that the results shown in Figure 1(b) appear upon usage.

```
#include <stdio.h>

int main(){
    int guess, attempts, secret = 24;

    printf("Guess the number in 3 attempts to win!\n\n");

    //Complete the rest of the program below

    return 0;
}
```

Figure 1(a): Guess-the-number game.

```
Guess the number in 3 attempts and win a prize!

Attempt #1: 65
Too big
Attempt #2: 24
You guessed it!
```

```
Guess the number in 3 attempts and win a prize!

Attempt #1: 90
Too big
Attempt #2: 10
Too small
Attempt #3: 31
Too big
You lost!
```

Figure 1(b): **Bold and underlined** denote user's input.

[10 marks]

Continued...

(b) Assume that `str1` contains the string "`Super`", `str2` contains the string "`Bat`", and `str3` contains the string "`man`". Assume the "skeleton" is already present and the string variables have already been initialised. Write the code snippets according to the instructions below.

(i) Just by using `printf` and the appropriate format specifier (e.g. `%d`), write the code that produces the following output from the variables (i.e. `str1`, `str2`, and/or `str3`):

```
Batman
```

[2 marks]

(ii) Using `printf` and the appropriate format specifier (e.g. `%d`) again, **select individual characters** from `str1`, `str2`, and/or `str3` to form the new word shown below:

```
Burp
```

[2 marks]

(iii) Concatenate the appropriate strings together by using a suitable **string function** to form the following word:

```
Superman
```

[2 marks]

(iv) Determine the length of `str1` by using a suitable **string function.**

[2 marks]

(v) A new string variable called `duplicate` is initialised. Write the code to copy `str2` to `duplicate`.

[2 marks]

Continued...

(c) Provide the output of the program in Figure 1(c) at the following lines:

```
1  int main(){
2      int a[5] = {10, 20, 30, 40, 50};
3
4      printf("%d ", *a);
5      printf("%d ", *a+1);
6      printf("%d ", *(a+1));
7      printf("%d ", a[4]);
8      printf("%d ", *(a+3)+4);
9
10     return 0;
11 }
```

Figure 1(c)

(i) Line 4
(ii) Line 5
(iii) Line 6
(iv) Line 7
(v) Line 8

[5 marks]

Continued...

Question 2

(a) Figure 2(a) shows an incomplete C program while Figure 2(b) shows its output. Read the program carefully and answer the following questions.

```
#include <stdio.h>

float ounce2pound(float ounce);
void pound2kg(float pound, float *kg);

int main()
{
    float ounce, pound, kg;
    printf("Please input the mass in ounce: ");
    scanf("%f", ounce);

    // Question Q2(a)(iii)
    printf("%.2f ounces is equivalent to %.2f
           pounds.\n", ounce, pound);

    // Question Q2(a)(iv)
    printf("%.2f pounds is equivalent to %.2f
           kilograms.\n", pound, kg);

    return 0;
}
```

Figure 2(a): Program for mass conversion.

```
Please input the mass in ounce: 200
200.00 ounces is equivalent to 12.50 pounds.
12.50 pounds is equivalent to 5.67 kilograms.
```

Figure 2(b): **Bold and underlined** denote user's input.

(i) Fill in the blanks for the following statement:

Function `ounce2pound()` takes in _____ number of input argument(s) and returns the result of _____ data type.

[2 marks]

(ii) Fill in the blanks for the following statement:

Function `pound2kg()` takes in _____ number of input argument(s) and returns _____ number of results.

[2 marks]

Continued...

(iii) Write the statement to call function `ounce2pound()` to convert the input mass from ounce to pound. Note: Do not declare any new variable.

[2 marks]

(iv) Write the statement to call function `pound2kg()` to convert the mass from pound to kg. Note: Do not declare any new variable.

[2 marks]

(v) Write the function definition of `ounce2pound()` which converts the floating point input parameter `ounce` and returns the pound as a result based on the ratio:1 pound = 16 ounces.

[4 marks]

(vi) Write the function definition of `pound2kg()` which takes in the floating point input parameter `pound` as well as a pointer pointing to the original variable `kg` from `main()`. It calculates and assigns the number of kilograms into the pointer `kg` based on the ratio:1 pound = 0.4536 kg

[4 marks]

(b) Identify **NINE (9)** lines with errors in the code listed in Figure 2(c). For each line with error, write down the line number followed by the corrected version in your answer. For example:

Line 1: `#include <stdio.h>`

The program stores 10 letters from the user into an array and subsequently, calculates the number of letter `A` in the array.

Hint: Assume that Line 3 with the prototype declaration is **correct**.

```
1  #include <stdio.c> //Ignore this. There are 9 others.
2
3  int count_A(char letter[]);
4
5  int main(){
6      int i, count;
7      char letter[10];
8
9      for(i=0; i<10; j++){
10         printf("letter[%i]: ", i);
11         scanf(" %c", &letter);
12     }
13
14     count_A = count_A(letter);
15
16     return 0;
17 }
```

Continued...

```
18
19  int countA(char letter[]){
20      int i, j=10;
21
22      for (i=0; i>10; i++){
23          if (letter[i] == A)
24              j++;
25      }
26
27      return A;
28  }
```

Figure 2(c): Source code with errors.

[9 marks]

Continued...

Question 3

(a) Answer **True** or **False** for the following statements:

(i) The `fgetc()` function returns NULL when the end of file is reached.
[1 mark]

(ii) The `fopen()` function returns NULL when it is not able to open a file.
[1 mark]

(iii) The `fgets()` function is used to write a string to a file.
[1 mark]

(b) Consider the `struct` definition and variables shown in Figure 3(a) below.

```
struct Flight
{
    char destination[50];
    int flight_id;
    float price;
};

struct Flight flight1, *flight2;
```

Figure 3(a): `struct` definition.

Identify the data types of the following variables:

(i) `flight1;`
[2 marks]

(ii) `flight2;`
[2 marks]

(iii) `flight2->destination;`
[2 marks]

(iv) `flight1.flight_id;`
[2 marks]

(c) Figure 3(b) shows an incomplete program that stores the calories of different food. Complete the program by answering the questions listed below.

Table 3(a): Calories in 100g of different types of food

Type of Food	Carb	Protein	Fat
Potato	65.5	8.4	0.9
Beef	0	110.5	125.3
Apple	5.8	1.4	2.0
Egg	4.6	55.7	98.2

Continued...

```
#include <stdio.h>

//Answer for Part (i)
//struct Calorie definition

int main(){

    int i;

    //Answer for Part (ii)
    //Declare and initialize an array called food

    //Answer for Part (iii)
    //Display the total of calories

    return 0;
}
```

Figure 3(b). Incomplete program.

(i) Define a suitable structure called `Calorie` that can be used to store the types of food and the corresponding values of carbohydrate, protein, and fat.

[5 marks]

(ii) Declare an array called `food` of the type `struct Calorie` and initialise it with the data in Table 3(a).

[5 marks]

(iii) Using a `for` loop, display the total calories for each type of food on the screen. The total calories is the summation of the calories contributed by each nutrient (i.e. carb, protein, fat). The sample output is as shown below.

```
Food        Total Calories
-----------------------------
Potato      74.8
Beef        235.80
Apple       9.20
Egg         158.50
```

[4 marks]

Continued...

Question 4

(a) Convert the following pseudocodes to their equivalent C statements. The expected answers should be like Figure 4(a).

Pseudocode:	If `speed` is greater than 100
C statement:	`if(speed > 100)`

Figure 4(a)

(i) If `item` is not 20 [1 mark]
(ii) If `grade` is either A or B [2 marks]
(iii) If `height` is less than 2 and greater than or equal to 1.2 [2 marks]
(iv) If `year` is odd [2 marks]

(b) Figure 4(b) shows a sequence of numbers.

11	22	44	88	176

Figure 4(b): Program output.

Write the C program segment that prints the sequence above using the following loops:

(i) `for` loop [3 marks]
(ii) `do-while` loop [3 marks]
(iii) `while` loop [3 marks]

(c) Write a complete C program that calculates and analyses the profit made by a self-service laundry shop. The shop has 5 washing machines, and they accept coin tokens worth RM1.50 each. The program should (i) prompt the user to key in number of tokens collected by each of the 5 machines **using a loop**, (ii) convert the tokens to profit (RM), (iii) calculate the total profit from all machines, and (iv) calculate the average profit. The results should be displayed in a table as shown in Figure 4(c).

To achieve the **FOUR (4)** objectives, you can use the following suggestions (not compulsory).

Continued...

(i) Using a loop, the program prompts the user to key in the number of tokens for 5 machines and **stores them in an array.**

(ii) Convert the value of the tokens to RM by multiplying with 1.5.

(iii) Using a loop, sum up all 5 machines' profit.

(iv) Calculate the average by dividing the total profit by 5.

```
Tokens for Machine 1: 2
Tokens for Machine 2: 4
Tokens for Machine 3: 6
Tokens for Machine 4: 8
Tokens for Machine 5: 10

MACHINE TOKENS  PROFIT (RM)
1       2       3.00
2       4       6.00
3       6       9.00
4       8       12.00
5       10      15.00

Total profit: RM45.00
Average profit per machine: RM9.00
```

Figure 4(c): **Bold and underlined** denote user's input.

[9 marks]

End of Paper